Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Semester Report

IoT Kitchen

Submitted to:
Annie Lane
262-444-3207
aelane@bu.edu

by

Team 21
IoT Kitchen

Team Members
Addison Dolido addison@bu.edu
Erin Dorsey edorsey1@bu.edu
Saransh Kothari saranshk@bu.edu
Yuran Shi yuran@bu.edu
Kenny Zheng ykzheng2@bu.edu

Submitted: December 10th, 2019

Table of Contents

Executive Summary........ 1
1.0 Introduction........ 2
2.0 Concept Development........ 4
3.0 System Description........ 7
4.0 First Semester Progress........ 11
5.0 Technical Plan........ 13
6.0 Budget Estimate........ 16
7.0 Attachments........ 17
7.1 Appendix 1 – Engineering Requirements........ 17
7.2 Appendix 2 – Gantt Chart........ 18
7.3 Appendix 3 – Technical Resources........ 19
7.4 Appendix 4 – Team Information........ 20

Executive Summary

Team 21 - IoT Kitchen

Many amateur chefs often face difficulty when trying to follow along with recipes and updating recipes to their preference in real time. In addition, there are many existing technologies to aid in cooking but none that concentrate them all into an easy to use hub. IoT Kitchen seeks to solve these issues as the ultimate smart cooking assistant for amateur home chefs. IoT Kitchen provides hands free assistance so that chefs can follow recipes as well as update recipes to their own preferences in real time. This is achieved with the help of a user friendly app interface, Natural Language Understanding (NLU) assistant, as well as several IoT sensors including a scale, thermometer, and measuring cups. The Android app interface will allow users to keep a record of their recipes made on their profile page, and find new recipes by viewing other users' profiles. If a user fails to find inspiration from their friends, they can search our curated recipe database with strategic recipe keywords such as ingredient or dietary restriction. When a user finds a recipe, they can initialize auditory recipe instructions through the NLU assistant. This assistant will help a user step through a recipe or jump to a specific ingredient or instruction. If the user comes across an element of the recipe that they'd like to change, they can vocalize this change to the NLU assistant, which will send that information back to the recipe database and app. Throughout cooking, the IoT sensors will track cooking time, temperature, and ingredients added, providing feedback to the app to update the recipe and to correct a user if they are following the recipe wrong. Combined, IoT Kitchen will make following along to recipe and cultivating new ones easier than ever before.

1.0 Introduction

If you are like most people, you do not look forward to the cooking process. Many amateur chefs often face difficulty when trying to follow along with recipes and updating recipes to their preference in real time. Although there are many existing technologies to aid in cooking, none concentrate specifically all into an easy to use hub. This is why IoT Kitchen was created. With smart kitchen appliances, you will save time in the kitchen. These futuristic appliances explain the steps of the cooking process, analyze cooking preferences, and make food preparation much more efficient.

IoT Kitchen is a smart kitchen assistant that utilizes an IoT scale as well as natural language processing to breakdown the barriers for amateur cooks. This application makes it easier than ever for amateur homecooks to make recipes by using our scale that sends data to our mobile application, sorting out what ingredients they are putting in. This alongside the natural language processing assistant that gives the user instructions as they are following it making cooking easier than ever.

Our approach to create IoT Kitchen consists of the following components: Hardware, Natural Language Understanding (NLU), and Mobile Application. In our hardware portion of this project, we plan on creating a scale, thermometer, and RFID measuring cups which will all relay information to the mobile application. Our mobile application recipe categories will be catered to chefs based upon their diet. For instance, if a user wishes to find dishes with chicken, then there will be a category specifically for chicken related recipes. This nifty device displays visual recipes in a step-by-step manner and helps you measure the perfect amounts for your catered recipes. If you decide to alter the recipe or if you do not have an ingredient, this gadget will adjust the other ingredients as appropriate or provide alternative choices for suitable substitutes. Lastly, if a user wishes to change a specific ingredient, they are able to do so with NLU assistant which will send that information back to the recipe database and mobile application. The IoT sensors will track cooking time, temperature and ingredients added all while providing feedback to the mobile application to update the recipe and correct a user if they are following the recipe incorrectly.

Each of the smart kitchen appliances above facilitates cooking, making it that much easier for the average person to create tasty dishes in surprisingly little time. Perhaps most important, consuming more nutritious food prepared in a smart kitchen will encourage people to eat healthy homemade dishes as opposed to unhealthy fast foods. This can not only shorten the cooking process but make people better at preparing nutrient dense meals.

Figure 1. Concept Drawing of IoT Kitchen Implemented in a Kitchen

2.0 Concept Development

In the 21st century, people are living in a faster pace than ever. As a consequence, people are spending less time in the kitchen since it is a time consuming task and without sufficient cooking experience, it could be a mess. The members of Team 21 can relate to the desire to cook but not have adequate experience to execute the recipes we'd like to achieve. Additionally, we found research conducted by the New York Times that shows some main concerns of amateur chefs:

1. Hard to follow the recipe
2. Tidiness of the kitchen
3. Health concern

From the research above, amateurs are concerned mostly about lack of instruction, difficulties when it comes to making a delicious dish, and making recipes that work with their health goals. IoT Kitchen brings a centralized solution to the market that addresses these difficulties that amatuer home chefs face. Our understanding is that in order to encourage more home cooks, we need to build a mobile app that has a user friendly interface and boost chefs' confidence when cooking. While there are many other devices that aid in certain aspects of cooking, we decided to approach these problems with one tightly integrated and easy to use system.

The first way that IoT Kitchen addresses the problems of amateur chefs is our recipe interface. This is fueled by a meal plan API that is connected to extract recipes and store them into the Google Firebase database. The API provides tasty recipes that are easily integrated with our Natural Language Understanding assistant with data such as total calories and macros. This data lends to a solution to health concerns by providing users will easy to track health information for all their favorite recipes.

To avoid the mess of having a dirty recipe book laying in the kitchen, the Natural Language Understanding (NLU) system will be implemented so that chefs can use instructions such as "next instruction" to move on to the next step instead of touching their recipe book or mobile phone with dirty hands. This NLU assistant ideally with have an accuracy rate of at least

90% in the English language. In the future it is possible to implement this system in more languages, but given who our client and audience are we chose to stick with English.

The Android app will also have the feature of displaying the amount of ingredient that has been added to the measuring bowl, making it easier than ever to follow along to a recipe. We considered just displaying the recipe on the app and not providing verbal instructions but we found that the combination of the two is particularly helpful and accommodates all types of chefs of different preferences and abilities.

To further increase the ease at which users can complete recipes, we decided to implement a series of IoT kitchen sensors. We chose to implement both a scale and measuring cups because this accounts for most recipes which use either weight or volume to specify ingredient amounts. We chose also to include a temperature sensor so that a user has data on all aspects of the cooking process, even during their inactive time. In addition, our client specified that she likes to see a lot of data, and we felt that temperature over time has a lot of potential for pleasing data visualizations. In order to be suitable for a kitchen environment, all of these sensors must be resistant to liquid spills. In addition, the temperature sensor must be able to function while in an oven, or up to 575°F.

To achieve real time updates from the sensors to the app, we decided to implement bluetooth features on both the Android app and scale which uses HC-05 for bluetooth connection. Since our app is capable of receiving data wireless from the scale, the front end will prompt user when desired amount of ingredient is being added, and it could move on to the next instruction. Similar functionality will be implemented on the temperature sensor and RFID cups to achieve real time updating. Combining that with the NLU function, the bluetooth will play a big role in achieving overall project success.

Last but not least is the database, which has prestored recipes that allow for user modification. This is significant because it allows users not only to follow recipes, but to customize them to make them their own. In addition, the database will be able to record the user's daily intake of macros and notify the user if their intake exceed the threshold. This database holds the whole Android app together as we translate user's voice control command into string and use that to extract corresponding information stored from database.

In all, we aim for IoT Kitchen to be the one stop shop for amateur chefs who want to follow along with healthy delicious recipes without making a total mess. Our easy to use hub and integrated sensors and Natural Language process provides a system that is easy to use and will hopefully motivate more and more people to get into the kitchen.

3.0 System Description

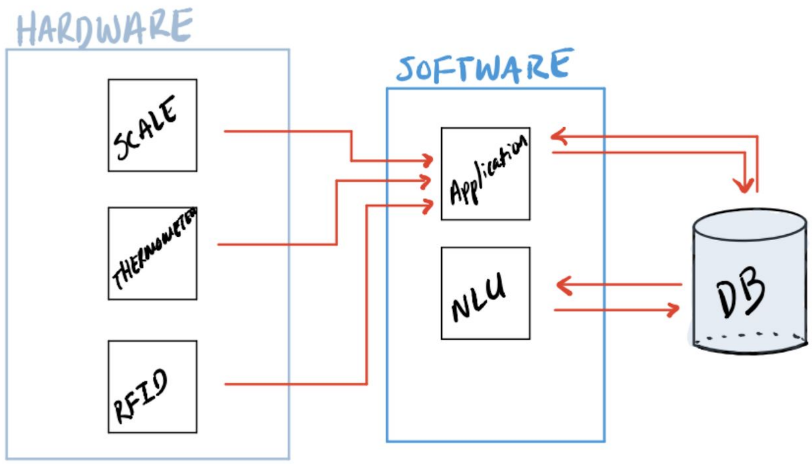

Figure 2. Whole System Block Diagram
The three hardware sensors send data to the software application that, alongside the Natural Language Understanding software module, reads and writes from the database

<u>Hardware</u>

The hardware elements for this project are various sensors that collect data about a user's cooking session. Initially there will be three unique sensors, two of which will transmit measured data to the android app via bluetooth. The data these sensors collect will be viewable by the cook to see more detailed information about the recipe they created, such as the weight of flour used or the temperature of the chicken versus time. These three sensors are described individually below.

The first sensor is a smart scale, that will measure the weight of ingredients used. This scale is made up of an arduino uno, a 5kg strain gauge load cell, an HX711 load cell amplifier, an HC-05 bluetooth module, an LCD screen, and two buttons. All of this will be contained in a water resistant 3D printed housing to allow for splashing and cleaning. The cook will be able to place a bowl onto the scale and press a button to tare the scale. Then when the instructions says to add 50 grams of flour, the scale can measure that the cook actually used 53 grams of flour. When the cook is done adding that ingredient they can push another button to send that data to the Android app and continue onto the next recipe instructions. The scale will be calibrated to measure to the nearest gram and will be sensitive enough to measure 1 gram.

The next sensor is an oven temperature sensor that will measure the internal temperature of food inside the oven, such as a chicken. This is made up of a type-k glass braid insulated stainless steel tip that can measure up to 700°C and is food and oven safe which is then connected to a MAX31856 thermocouple amplifier that works with temperatures well above 700°C. All of this is also attached to an arduino uno and an HC-05 bluetooth module. The cook will be able to stick the thermocouple into the food item going into the oven and then attach the rest of the components onto the outside of the oven, so that the bluetooth signal receives no interference and the hardware does not overheat. This will allow the cook to tell when their food has reached the proper internal temperature and is ready to be removed from the oven.

Finally, we will have RFID tags on several appliances, such as measuring cups and knives to record what tools are used for certain tasks and the volume of ingredients used. We will create the RFID tags by combining an RFID chip, antenna and encapsulating it within a substrate of our choice. These tags will be passive and will function in the high frequency range (13.56 MHz) so that an android smartphone may used as an RFID reader by utilizing its NFC protocol. The antennas used will consist of either a copper or aluminium coil with 3-7 turns. As a result of functioning in the high frequency range, these tags will have a read range of less than 3 feet. This will aid the user in ensuring that an RFID tag is scanned only when intended.

Software Mobile Application

The software component of this project acts as a hub that connects to the hardware components HC-05 module via bluetooth and Wifi, displaying data on the tip of your hands. This portion of the project is what allows users to store and update recipes and search for recipes based upon different preferences. Our main sections of the mobile application includes the following: Home, Profile, Recipes, and Settings all using Android Studio, Java and Google FIrebase.

In the Home section, users are presented with the newsfeed which will include suggestions based upon the time and/or day. For instance, if it is the morning, there will be a recipe suggestions such as omelets with bacon. Following the Home page will be the users very own profile page which will include favorite recipes, things they have cooked, calendar, saved recipes and total time it took them to cook for that specific day.

The Profile page consists of user authentication which is for users to login to their specific data by using Google Firebase. Users are able to display their favorite dishes as well as dishes they have previously cooked.

The Recipes page is catered based upon the users diet and will include the following: chicken, beef, low-carb, high-protein, vegan, etc which is called from the MealDB API. By doing so, users are able to search and filter out what recipes work for them. This will then navigate them to the preview recipes page before they start cooking which includes ingredients, recipe instructions, weight and amount. Once the user wishes to start cooking, they will be shown a page of instructions which will help guide them through each instruction through the utilization of our NLU. Lastly, we plan on including a settings tab that can sync and configure devices such as the Google Home. This is also where the bluetooth connectivity with the Arduino UNO is made possible via Bluetooth.

Figure 3. Preliminary GUI of Mobile Application

Natural Language Understanding

The Natural Language Understanding (NLU) elements of the project allow a user to follow and interact with a recipe through voice commands and auditory responses. The user can interact with the NLU agent in the main app or through a Google Smart device such as an Android phone enabled with Google Assistant or a Google Home. The agent works by receiving voice commands from a user and using Google's Dialogflow NLU API it can match specific words and commands to an intent. Intents correlate to specific functions and processes that the agents will execute. Additional specificity comes in the form of entities, which are specific variables for intent functions that specific keywords are matched to. For example, the word "first" is matched to the stepNum entity for the intent to find a specific instruction or ingredient. Then, fulfillment code written in javascript is used to connect the Dialogflow agent to a Google Firebase database containing recipe information. This code connects the agent to the database via a webhook request and either writes or reads to the database in a different function for each intent. This fulfillment returns parts of the recipe to Dialogflow for the agent to speak aloud depending on the input of the user.

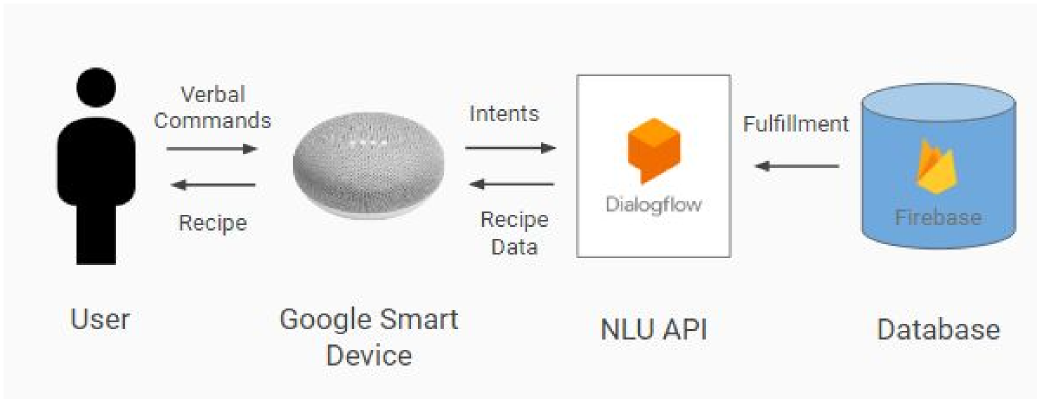

Figure 4. Block Diagram of NLU Agent
A user speaks a command to the Google Smart Device which is interpreted by the Dialoglow NLU API. Data is retrieved from Firebase via fulfillment and recipe data is conveyed to user.

The functionality of this NLU agent is twofold. A user can utilize this agent to follow along with a recipe by asking for ingredients or instructions, or a user can instruct the agent to change elements of a recipe. First, a user initializes the agent by saying "Talk to IoT Kitchen" to whatever Google smart device they prefer to use. Then they specify the code of whatever recipe they would like to make. This initializes the agent to pull from the database page of that recipe. To follow a recipe, a user can ask for the title, details (servings and time information), ingredients, or instructions. They can specify which ingredient or instruction they would like, such as "first", "second", "last", etc.. Alternatively, the user can step through the recipe with commands like "next", "back", or "repeat". These queries are matched to fields in the recipe database page and returned to the user. If the user would like to change an element of the recipe at any point they must specify whether they would like to change an ingredient, instruction, or detail, and which number they would like to change. These specifications will match the user command to a key in the recipe database page. Then, the user can supply the alteration to the recipe text and the agent will send a script to automatically update that value in the recipe database with the new text. This saves all changes that a chef makes in real time.

4.0 First Semester Progress

Hardware

The main hardware focus for this semester was to gain a working scale prototype that could transmit data to an android app. The scale prototype created consists of an arduino uno, a 5kg strain gauge load cell, an HX711 load cell amplifier, an HC-05 bluetooth module, and an LCD screen. The load cell measures the tension and compression caused by the weight on top of it. Thus, a simple wooden housing was created to allow for strain on the load cell. Once that was built and the circuitry was put together the scale was calibrated.

Calibration was accomplished with a set of known weights, an arduino script, and several calculations. The load cell's output changes linearly in proportion to the weight on top of it. So calibration consisted of finding the best slope and y-intercept for this configuration.

The next step was to then set up the HC-05 bluetooth module, configuring it's name and password and then ensuring that the android app would connect and receive data from it. The app now automatically connects to the scale, requesting the password and successfully receives data sent from the scale.

The scale also went through several tests. It measured all weights properly to the nearest gram, was sensitive enough to measure 1 gram, and was able to be tared. Additionally, after taring the scale was no less accurate than before taring.

Finally, near the end of the semester, focus shifted towards the temperature sensor. Components that met our requirements were picked and ordered: a type-k glass braid insulated stainless steel tip and a MAX31856 thermocouple amplifier. A circuit schematic was also created and as soon as the parts arrive a temperature sensor prototype will be assembled.

Software - Mobile Application

This semester our primary focus was towards the backend of the mobile application. Our progress includes user authentication with Firebase, Bluetooth connectivity with the scale and recipe categories of retrieving meal recipes by utilizing and calling the MealDB API.

In terms of authentication, we were able to log in users through Google Firebase as well as the option to sign out of the mobile application and log back in with a different user. Each time a new user login, the program will load the user a copy of recipe from our origin database and put under their profile, so they will be access and modify recipe.Once logged in, the user is able to connect to the Arduino via Bluetooth which will display the scale data onto the mobile application. Since our app is capable of receiving data wireless from the scale, the front end will prompt user when desired amount of ingredient is being added, and it could move on to the next instruction.

In the recipe portion of this mobile application, it is ensuring that the categories of recipes based upon different options such as chicken, beef, etc. This was done through calling the meal database from MealDB API which included information such as image, instructions, and ingredients. A demonstration of how this would work is if a user wishes to find recipes geared towards vegan dishes, they navigate to the vegan options which will then show them vegan dishes that they can make. This also gives them the option of viewing the dishes on YouTube for visual learners as well the country of origin and instructions for the specific dish. Finally, there was some progress in the search functionality, favorite tab and front end of the mobile application for the food recipes. This included an image and category based upon each selection and specific recipes in which the user favorited.

Natural Language Understanding

The first element of progress for natural language understanding (NLU) this semester was building a proof-of-concept Android app with a module that converts text to speech. Built in packages were used to write an app with would convert recipe text into speech. This module will be used when the Dialogflow agent is fully integrated with the main app.

The majority of progress this semester has been made building an NLU agent using Google's Dialogflow API. The Dialogflow development console was used to build intents and entities so that the verbal commands from the user can be interpreted into function calls to the database. This semester intents were made to get the recipe title, details, and instructions an ingredients as specified by the stepNum entity. This allows a user to ask for a specific instruction or ingredient or step through. The agent can also match a user's command for a specific recipe with the recipeTitle entity. During first prototype testing, all user commands were correctly matched to the appropriate text with 100% accuracy. The agent also did not crash when posed with a voice command that it could not map to an intent.

The next element of progress for the NLU this semester was linking the Dialogflow agent to a database via fulfillment. The database was built using Google Firebase and was initialized with two simple recipes used for testing. Fulfillment code was written in javascript with a function corresponding to each intent, with additional functions to set the recipe database page and test reading and writing from the database. This code was successful in pulling the correct recipe data for each intent with 100% accuracy during first prototype testing. It also succeeded in not crashing when given input that asked for data that was not present in the database.

Finally, there was some progress made in the deployment of this NLU agent to a Google Smart device. A Google Home Mini was acquired and configured for use on the school network. A Google Assistant Action was built, fueled by the Dialogflow Agent. The deployment process will be completed in the next semester.

5.0 Technical Plan

Hardware - Scale

Task 1. Decrease circuitry volume
All of the parts making up the scale will be soldered together to ensure stability and decrease the volume that they take up. If necessary a PCB may be built to replace some of the parts to further decrease volume and decrease power consumption.
Lead: Addison

Task 2. Create a water resistant housing
A 3D printed casing for the internal circuitry of the scale will be designed and built to ensure the scale is water resistant. The secondary goal of the casing will be to increase the overall aesthetics and useability of the scale.
Lead: Addison

Hardware - Temperature Sensor

Task 1. Build a temperature sensor prototype
The parts of the temperature sensor prototype (Arduino Uno, Thermocouple, Amplifier, HC-05) will be put together according to the schematic already created. This will then be calibrated to report the temperature in Farenheit.
Lead: Addison

Task 2. Test the temperature sensor prototype in the oven
The temperature sensor will be placed in the oven to ensure that it both survives and accurately reads the required temperatures. The feasibility of having the wire of the sensor go through the oven door will also be explored, ensuring no significant heat is lost from the oven.
Lead: Addison

Task 3. Temperature sensor housing created
A housing for the temperature sensor will be designed and built to withstand oven temperatures and be food safe.
Lead: Addison

Task 4. Temperature sensor tested in food items
The temperature sensor with it's housing will be tested by measuring the internal temperatures of different food items throughout baking. The reported temperature will then be compared to that of a commercial food thermometer. The ease of use in getting the sensor in and out of the food will also be noted.
Lead: Addison

Task 5. Ensure temperature sensor is water resistant
The circuitry of the temperature sensor will be soldered to decrease volume and a PCB will be created if necessary to further decrease volume and power consumption if necessary. A housing will then be created to ensure that the hardware is water resistant and increase both ease of use and aesthetic.
Lead: Addison

Hardware - RFID

Task 1. Design Antennas
Antennas will be constructed of either a copper or aluminum coil with approximately 3-7 turns. It will be made so that the RFID tags may function in the high frequency range (13.56MHz).
Lead: Saransh

Task 2. Combine RFID chip, Antenna, and Substrate
The antennas will be deposited on a flexible substrate of our choice which will most likely be plastic. The RFID chip will then be connected to the antenna on the substrate itself.
Lead: Saransh

Task 3. Checking Readability of RFID tags
An android smartphone will be used as the RFID reader within the created RFID system. The phone's NFC protocol will be utilized to power the RFID tags within the high frequency range. We will check the readability of the passive RFID tags within less than 3 feet of the reader. If the android smartphone successfully reads the tags, it will be confirmation that the RFID tags are operating at 13.56MHz.
Lead: Saransh

Task 4. Integrating RFID tags with Kitchen Equipment
The constructed RFID tags will be integrated with measuring cups and other kitchen appliances. The goal is to combine the tags with the kitchen appliances in such a fashion that allows maximum convenience for the user when scanning a particular tag. Furthermore, it will be imperative to ensure that the RFID tags are protected as the kitchen appliances will most definitely be exposed to liquids.
Lead: Saransh

Software

Task 1. Front End of Mobile Application
An aesthetic user interface using Adobe XD and Sketch for UI/UX design. This is to amplify the user friendly experience through animations, and aesthetic of the mobile application for recipes, user profile, etc..
Lead: Kenny

Task 2. Back End of Mobile Application
Our app should be able to recognize the desired part to put into the database. The program needs to be capable of recognizing the ingredients and amount of the ingredient and only save those into the Google

Firebase Database. Temperature sensor and RFID cup feature will be incorporated later on using bluetooth.
Lead: Yuran

Task 3. Meal Database & API
Integrate more meal recipes into Google Firebase from meal database APIs. Users should be able to edit their recipes which will be stored on their user profile.
Lead Kenny Assisting: Yuran

Natural Language Understanding

Task 1. Write Privacy Policy for Action
A privacy policy for IoT Kitchen Google Assistant Action that is compliant with Actions on Google policies and procedures will be designed and written. Task will be achieved when privacy policy is submitted with Action and Action is approved for deployment.
Lead: Erin

Task 2. Build Intents for User Changes
Dialogflow intents that match user commands to change recipes will be designed, built, and tested. Fulfillment code will be written to write user changes to database. Fulfillment will be timed for "real-time" updating and should achieve this in under three seconds. Intents should be able to match user commands with 90% accuracy.
Lead: Erin

Task 3. Integrate NLU Agent
Using the Dialogflow V2 API, NLU functionality will be integrated directly into the main app for users who do not have access to a Google smart device. This task will be achieved when functionality through main app is equivalent for each intent to functionality of device based agent. Equivalency will be evaluated with accuracy of matching and ability to not crash.
Lead: Erin Assisting: Kenny, Yuran

Task 4. Improve Usability of Integrated Agent
An aesthetic user interface will be created to improve the usability and user experience of the app-integrate agent. This interface will have a design similar to the rest of the app to allow for continuity. Users should not be able to distinguish between the aesthetic of the rest of the app and this integrated agent.
Lead: Erin

6.0 Budget Estimate

In this section you discuss WHAT ARE THE BUDGET IMPLICATIONS AND CONSTRAINTS for your effort.

Item	Description	Cost
1	Google - Home Mini (1st Generation) - Smart Speaker with Google Assistant ***Donated**	$20.00
2	GL.iNET GL-MT300N-V2 Mini Travel Router ***Donated**	$20.50
3	Sketch 1 Year License - Student Discount	$52.84
4	NFC ANTENNA FOR IOT - Abracon LLC	$3.20
5	MIFARE Classic 1K Dia 25mm adhesive RFID Sticker NFC tag	$14.98
6	Adafruit Micro NFC/RFID Transponder - NTAG203 13.56MHz [ADA2800]	$6.21
7	Arduino Uno	$18.50
8	HC-05 Bluetooth Module	$4.99
9	5kg Strain Gauge Load Cell & HX711 Load Cell Amplifier	$14.95
10	16x2 LCD Display	$6.22
11	Thermocouple Type-K Glass Braid Insulated Stainless Steel Tip	$9.95
12	MAX31856 Universal Thermocouple Amplifier	$17.50
	Total	$189.84

7.0 Attachments

7.1 Appendix 1 – Engineering Requirements

Team 21 Team Name: IoT Kitchen

Requirement	Value, range, tolerance, units
Natural Language Understanding	English language understanding with 90% accuracy rate
Android Mobile Device	Android supported device (Lollipop and above)
Bluetooth Connection	Stable bluetooth connection in a 5 meter range
Scale Sensitivity	Can measure a change of 1 gram
Scale Precision	Measures accurately to the nearest gram
Temperature Sensor Precision	Measures accurately to the nearest 1°C (1.8°F)
Temperature Sensor Heat Resistance and Range	Can withstand and measure temperatures ranging from 0°C up to 300°C (32°F - 575°F)
RFID Tag Range	All RFID tags should be readable when they are within 3 feet and directly in the line of sight of the reader
Spill Resistance	The scale, temperature sensor, and RFID cups must be resistant to water so that none of the electronics will fail when splashed with water

7.2 Appendix 2 – Gantt Chart

7.3 Appendix 3 – Technical Reference

[1] "Dialogflow Documentation," Google, Mountain View, CA, USA, October 20, 2019 [Online]. https://cloud.google.com/dialogflow/docs/

[2] S. Al-Mutlaq and Alex, *Load Cell Amplifier HX711 Breakout Hookup Guide*, Sparkfun, July 22, 2016 [Online]. https://www.sparkfun.com/search/results?term=load+cell+amplifier

[3] Avia Semiconductor, "24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales" HX711 datasheet.

[4] Maxim Integrated, "Precision Thermocouple to Digital Converter with Linearization" MAX31856 datasheet.

[5] Johnson, Felicity. "Firebase: User Sign Up, Login & Data Management." *Medium*, Medium, 9 Nov. 2016, [Online]. https://medium.com/@felicity.johnson.mail/firebase-user-sign-up-login-data-management-992d778b167

[6]"Android Bluetooth with Examples." *Tutlane*, [Online]. www.tutlane.com/tutorial/android/android-bluetooth-with-examples.

7.4 Appendix 4 – Team Information

Name	Bio	Team Role
Addison Dolido addison@bu.edu	Addison Dolido is a senior studying Electrical Engineering. Addison has always been curious as to how the technology around him works and decided to learn how to create new devices. He has a broad range of technical experience including software development in C#, Python, Java, and Javascript, 3D modeling and printing, hardware design and integration, and even artificial intelligence and machine learning. Addison's passion for exercise and love to cook drew him to the IoT Kitchen project, while his desire to broaden his technical knowledge further caused him to take the role as hardware lead.	Hardware
Erin Dorsey edorsey1@bu.edu	Erin Dorsey is a senior studying Computer Engineering who hails from Southbury, CT. Erin has experience in software development in C/C++, C#, Python, and Javascript, in addition to her significant experience in project management honed by various school projects and internships. At one of these internships she was exposed to the development process of Natural Language Understanding agents, inspiring her to take the lead on this element of the IoT Kitchen project. Outside of class, Erin is pursuing a career in cybersecurity consulting. In her free time, she is an avid cook and baker.	NLU
Saransh Kothari saranshk@bu.edu	Saransh Kothari is currently a junior pursuing a degree in Electrical Engineering. His interests lie within the fields of artificial intelligence and quantum computing. Saransh left Hong Kong to begin his educational journey in Boston and started in the Questrom School of Business. After a year, he transferred into the College of Engineering to garner a technical skill set to supplement his strong background in business. Saransh is an Entrepreneur at heart and plans to delve deeper down this path upon graduation. Saransh is known as an ardent traveler who has visited over 50 countries thus far.	Hardware
Yuran Shi yuran@bu.edu	Yuran Shi is a senior Computer Engineering student with a minor in Chemistry. Yuran has experience in software development in C/C++, C# and verilog. Yuran also gained experience of using database after internship at an insurance company. Yuran's experience with database and passion for cooking inspired him to take part in the back-end development for the IoT kitchen mobile application. Outside school, Yuran is an advanced sommelier and decide to continually pursue higher certificate.	Software
Kenny Zheng ykzheng2@bu.edu	Kenny Zheng is a senior studying Computer Engineering with a concentration in Technology Innovation. Kenny gained interest in software development after dropping out from college as a freshman. His experience lies in C/C++, Python, HTML, CSS and JavaScript after working as a web developer, and teaching assistant. Inspired by software development, Kenny decided to take his prior experience and knowledge in his role as the front-end developer for the IoT Kitchen mobile application. Besides schoolwork, Kenny enjoys sleeping, eating, and more sleep. Kenny also enjoys long walks while jamming to old Tay Tay on Spotify.	Software